INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC 1/SC 29/WG 11 N 3056
Maui, December 1999

ISO/IEC JTC 1/SC 29

Date: 2000-01-31

APPENDIX B

ISO/IEC 14496-2:1999/Amd.1:2000(E)

ISO/IEC JTC 1/SC 29/WG 11

SC 29 Secretariat: JISC

Information technology — Coding of audio-visual objects — Part 2: Visual

Amendment 1: Visual extensions

Technologies de l'information — Codage des objets audiovisuels — Partie 2 : Visuel

Document type: International Standard
Document subtype: Amendment
Document stage: (50) Approval
Document language: E

C:\My Documents\MPEG-4\w3056.doc STD Version 1.0

video_signal_type() {	No. of bits	Mnemonic
video_signal_type	1	bslbf
if (video_signal_type) {		
video_format	3	uimsbf
video_range	1	bslbf
colour_description	1	bslbf
if (colour_description) {		
colour_primaries	8	uimsbf
transfer_characteristics	8	uimsbf
matrix_coefficients	8	uimsbf
}		
}		
}		

6.2.2.1 User data

user_data() {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
while(next_bits() != '0000 0000 0000 0000 0000 0001') {		
user_data	8	uimsbf
}		
}		

6.2.3 Video Object Layer

VideoObjectLayer() {	No. of bits	Mnemonic
if(next_bits() == video_object_layer_start_code) {		
short_video_header = 0		
video_object_layer_start_code	32	bslbf
random_accessible_vol	1	bslbf
video_object_type_indication	8	uimsbf
is_object_layer_identifier	1	uimsbf
if (is_object_layer_identifier) {		
video_object_layer_verid	4	uimsbf
video_object_layer_priority	3	uimsbf
}		
aspect_ratio_info	4	uimsbf
if (aspect_ratio_info == "extended_PAR") {		
par_width	8	uimsbf
par_height	8	uimsbf
}		
vol_control_parameters	1	bslbf
if (vol_control_parameters) {		
chroma_format	2	uimsbf
low_delay	1	uimsbf
vbv_parameters	1	blsbf

if (vbv_parameters) {		
first_half_bit_rate	15	uimsbf
marker_bit	1	bslbf
latter_half_bit_rate	15	uimsbf
marker_bit	1	bslbf
first_half_vbv_buffer_size	15	uimsbf
marker_bit	1	bslbf
latter_half_vbv_buffer_size	3	uimsbf
first_half_vbv_occupancy	11	uimsbf
marker_bit	1	blsbf
latter_half_vbv_occupancy	15	uimsbf
marker_bit	1	blsbf
}		
}		
video_object_layer_shape	2	uimsbf
if (video_object_layer_shape == "grayscale" && video_object_layer_verid != '0001')		
video_object_layer_shape_extension	4	uimsbf
marker_bit	1	bslbf
vop_time_increment_resolution	16	uimsbf
marker_bit	1	bslbf
fixed_vop_rate	1	bslbf
if (fixed_vop_rate)		
fixed_vop_time_increment	1-16	uimsbf
if (video_object_layer_shape != "binary only") {		
if (video_object_layer_shape == "rectangular") {		
marker_bit	1	bslbf
video_object_layer_width	13	uimsbf
marker_bit	1	bslbf
video_object_layer_height	13	uimsbf
marker_bit	1	bslbf
}		
interlaced	1	bslbf
obmc_disable	1	bslbf
if (video_object_layer_verid == '0001')		
sprite_enable	1	bslbf
else		
sprite_enable	2	uimsbf
if (sprite_enable== "static" || sprite_enable == "GMC") {		
if (sprite_enable != "GMC") {		
sprite_width	13	uimsbf
marker_bit	1	bslbf
sprite_height	13	uimsbf
marker_bit	1	bslbf
sprite_left_coordinate	13	simsbf
marker_bit	1	bslbf
sprite_top_coordinate	13	simsbf
marker_bit	1	bslbf

}		
no_of_sprite_warping_points	6	uimsbf
sprite_warping_accuracy	2	uimsbf
sprite_brightness_change	1	bslbf
if (sprite_enable != "GMC")		
low_latency_sprite_enable	1	bslbf
}		
if (video_object_layer_verid != '0001' && video_object_layer_shape != "rectangular")		
sadct_disable	1	bslbf
not_8_bit	1	bslbf
if (not_8_ bit) {		
quant_precision	4	uimsbf
bits_per_pixel	4	uimsbf
}		
if (video_object_layer_shape=="grayscale") {		
no_gray_quant_update	1	bslbf
composition_method	1	bslbf
linear_composition	1	bslbf
}		
quant_type	1	bslbf
if (quant_type) {		
load_intra_quant_mat	1	bslbf
if (load_intra_quant_mat)		
intra_quant_mat	8*[2-64]	uimsbf
load_nonintra_quant_mat	1	bslbf
if (load_nonintra_quant_mat)		
nonintra_quant_mat	8*[2-64]	uimsbf
if(video_object_layer_shape=="grayscale") {		
for(i=0; i<aux_comp_count; i++) {		
load_intra_quant_mat_grayscale	1	bslbf
if(load_intra_quant_mat_grayscale)		
intra_quant_mat_grayscale[i]	8*[2-64]	uimsbf
load_nonintra_quant_mat_grayscale	1	bslbf
if(load_nonintra_quant_mat_grayscale)		
nonintra_quant_mat_grayscale[i]	8*[2-64]	uimsbf
}		
}		
}		
if (video_object_layer_verid != '0001')		
quarter_sample	1	bslbf
complexity_estimation_disable	1	bslbf
if (!complexity_estimation_disable)		
define_vop_complexity_estimation_header()		
resync_marker_disable	1	bslbf
data_partitioned	1	bslbf
if(data_partitioned)		
reversible_vlc	1	bslbf

if(video_object_layer_verid != '0001') {		
newpred_enable	1	bslbf
if (newpred_enable) {		
requested_upstream_message_type	2	uimsbf
newpred_segment_type	1	bslbf
}		
reduced_resolution_vop_enable	1	bslbf
}		
scalability	1	bslbf
if (scalability) {		
hierarchy_type	1	bslbf
ref_layer_id	4	uimsbf
ref_layer_sampling_direc	1	bslbf
hor_sampling_factor_n	5	uimsbf
hor_sampling_factor_m	5	uimsbf
vert_sampling_factor_n	5	uimsbf
vert_sampling_factor_m	5	uimsbf
enhancement_type	1	bslbf
if(video_object_layer == "binary" && hierarchy_type== '0') {		
use_ref_shape	1	bslbf
use_ref_texture	1	bslbf
shape_hor_sampling_factor_n	5	uimsbf
shape_hor_sampling_factor_m	5	uimsbf
shape_vert_sampling_factor_n	5	uimsbf
shape_vert_sampling_factor_m	5	uimsbf
}		
}		
}		
else {		
if(video_object_layer_verid !="0001") {		
scalability	1	bslbf
if(scalability) {		
shape_hor_sampling_factor_n	5	uimsbf
shape_hor_sampling_factor_m	5	uimsbf
shape_vert_sampling_factor_n	5	uimsbf
shape_vert_sampling_factor_m	5	uimsbf
}		
}		
resync_marker_disable	1	bslbf
}		
next_start_code()		
while (next_bits()== user_data_start_code){		
user_data()		
}		
if (sprite_enable == "static" && !low_latency_sprite_enable)		
VideoObjectPlane()		
do {		

if (next_bits() == group_of_vop_start_code)		
Group_of_VideoObjectPlane()		
VideoObjectPlane()		
} while ((next_bits() == group_of_vop_start_code) || (next_bits() == vop_start_code))		
} else {		
short_video_header = 1		
do {		
video_plane_with_short_header()		
} while(next_bits() == short_video_start_marker)		
}		
}		

define_vop_complexity_estimation_header() {	No. of bits	Mnemonic
estimation_method	2	uimsbf
if (estimation_method ='00' || estimation_method == '01') {		
shape_complexity_estimation_disable	1	
if (!shape_complexity_estimation_disable) {		bslbf
opaque	1	bslbf
transparent	1	bslbf
intra_cae	1	bslbf
inter_cae	1	bslbf
no_update	1	bslbf
upsampling	1	bslbf
}		
texture_complexity_estimation_set_1_disable	1	bslbf
if (!texture_complexity_estimation_set_1_disable) {		
intra_blocks	1	bslbf
inter_blocks	1	bslbf
inter4v_blocks	1	bslbf
not_coded_blocks	1	bslbf
}		
marker_bit	1	bslbf
texture_complexity_estimation_set_2_disable	1	bslbf
if (!texture_complexity_ estimation_set_2_disable) {		
dct_coefs	1	bslbf
dct_lines	1	bslbf
vlc_symbols	1	bslbf
vlc_bits	1	bslbf
}		
motion_compensation_complexity_disable	1	bslbf
If (!motion_compensation_complexity_disable) {		
apm	1	bslbf
npm	1	bslbf
interpolate_mc_q	1	bslbf
forw_back_mc_q	1	bslbf

halfpel2	1	bslbf
halfpel4	1	bslbf
}		
marker_bit	1	bslbf
if(estimation_method == '01') {		
version2_complexity_estimation_disable	1	bslbf
if (!version2_complexity_estimation_disable) {		
sadct	1	bslbf
quarterpel	1	bslbf
}		
}		
}		
}		

6.2.4 Group of Video Object Plane

Group_of_VideoObjectPlane() {	No. of bits	Mnemonic
group_of_vop_start_code	32	bslbf
time_code	18	
closed_gov	1	bslbf
broken_link	1	bslbf
next_start_code()		
while (next_bits()== user_data_start_code){		
user_data()		
}		
}		

6.2.5 Video Object Plane and Video Plane with Short Header

VideoObjectPlane() {	No. of bits	Mnemonic
vop_start_code	32	bslbf
vop_coding_type	2	uimsbf
do {		
modulo_time_base	1	bslbf
} while (modulo_time_base != '0')		
marker_bit	1	bslbf
vop_time_increment	1-16	uimsbf
marker_bit	1	bslbf
vop_coded	1	bslbf
if (vop_coded == '0') {		
next_start_code()		
return()		
}		

7	SMPTE 240M (1987) $E'_Y = 0,701\ E'_G + 0,087\ E'_B + 0,212\ E'_R$ $E'_{PB} = -0,384\ E'_G + 0,500\ E'_B - 0,116\ E'_R$ $E'_{PR} = -0,445\ E'_G - 0,055\ E'_B + 0,500\ E'_R$
8-255	reserved

In the case that video_signal_type() is not present in the bitstream or colour_description is zero the matrix coefficients are assumed to be those corresponding to matrix_coefficients having the value 1.

In the case that video_signal_type() is not present in the bitstream, video_range is assumed to have the value 0 (a range of Y from 16 to 235 for 8-bit video).

6.3.2.1 User data

user_data_start_code: The user_data_start_code is the bit string '000001B2' in hexadecimal. It identifies the beginning of user data. The user data continues until receipt of another start code.

user_data: This is an 8 bit integer, an arbitrary number of which may follow one another. User data is defined by users for their specific applications. In the series of consecutive user_data bytes there shall not be a string of 23 or more consecutive zero bits.

6.3.3 Video Object Layer

video_object_layer_start_code: The video_object_layer_start_code is a string of 32 bits. The first 28 bits are '0000 0000 0000 0000 0000 0001 0010' in binary and the last 4-bits represent one of the values in the range of '0000' to '1111' in binary. The video_object_layer_start_code marks a new video object layer.

video_object_layer_id: This is given by the last 4-bits of the video_object_layer_start_code. The video_object_layer_id uniquely identifies a video object layer.

short_video_header: The short_video_header is an internal flag which is set to 1 when an abbreviated header format is used for video content. This indicates video data which begins with a short_video_start_marker rather than a longer start code such as visual_object_ start_code. The short header format is included herein to provide forward compatibility with video codecs designed using the earlier video coding specification ITU-T Recommendation H.263. All decoders which support video objects shall support both header formats (short_video_header equal to 0 or 1) for the subset of video tools that is expressible in either form.

video_plane_with_short_header(): This is a syntax layer encapsulating a video plane which has only the limited set of capabilities available using the short header format.

random_accessible_vol: This flag may be set to "1" to indicate that every VOP in this VOL is individually decodable. If all of the VOPs in this VOL are intra-coded VOPs and some more conditions are satisfied then random_accessible_vol may be set to "1". The flag random_accessible_vol is not used by the decoding process. random_accessible_vol is intended to aid random access or editing capability. This shall be set to "0" if any of the VOPs in the VOL are non-intra coded or certain other conditions are not fulfilled.

video_object_type_indication: Constrains the following bitstream to use tools from the indicated object type only, e.g. Simple Object or Core Object, as shown in Table 6-10.

Table 6-10 -- FLC table for video_object_type indication

Video Object Type	Code
Reserved	00000000
Simple Object Type	00000001

Simple Scalable Object Type	00000010
Core Object Type	00000011
Main Object Type	00000100
N-bit Object Type	00000101
Basic Anim. 2D Texture	00000110
Anim. 2D Mesh	00000111
Simple Face	00001000
Still Scalable Texture	00001001
Advanced Real Time Simple	00001010
Core Scalable	00001011
Advanced Coding Efficiency	00001100
Advanced Scalable Texture	00001101
Simple FBA	00001110
Reserved	00001111 – 11111111

is_object_layer_identfier: This is a 1-bit code which when set to '1' indicates that version identification and priority is specified for the visual object layer. When set to '0', no version identification or priority needs to be specified.

video_object_layer_verid: This is a 4-bit code which identifies the version number of the video object layer. Its meaning is defined in Table 6-11. If both visual_object_verid and video_object_layer_verid exist, the semantics of video_object_layer_verid supersedes the other. When this field does not exist, the value of video_object_layer_verid is substituted by the value of visual_object_verid.

Table 6-11 -- Meaning of video_object_layer_verid

video_object_layer_verid	Meaning
0000	Reserved
0001	object type listed in Table 9-1
0010	object type listed in Table V2 - 39
0011 - 1111	Reserved

video_object_layer_priority: This is a 3-bit code which specifies the priority of the video object layer. It takes values between 1 and 7, with 1 representing the highest priority and 7, the lowest priority. The value of zero is reserved.

aspect_ratio_info: This is a four-bit integer which defines the value of pixel aspect ratio. Table 6-12 shows the meaning of the code. If aspect_ratio_info indicates extended PAR, pixel_aspect_ratio is represented by par_width and par_height. The par_width and par_height shall be relatively prime.

Table 6-12 -- Meaning of pixel aspect ratio

aspect_ratio_info	pixel aspect ratios
0000	Forbidden
0001	1:1 (Square)
0010	12:11 (625-type for 4:3 picture)
0011	10:11 (525-type for 4:3 picture)
0100	16:11 (625-type stretched for 16:9 picture)
0101	40:33 (525-type stretched for 16:9 picture)
0110-1110	Reserved

1111	extended PAR

par_width: This is an 8-bit unsigned integer which indicates the horizontal size of pixel aspect ratio. A zero value is forbidden.

par_height: This is an 8-bit unsigned integer which indicates the vertical size of pixel aspect ratio. A zero value is forbidden.

vol_control_parameters: This a one-bit flag which when set to '1' indicates presence of the following parameters: chroma_format, low_delay, and vbv_parameters.

chroma_format: This is a two bit integer indicating the chrominance format as defined in the Table 6-13.

Table 6-13 -- Meaning of chroma_format

chroma_format	Meaning
00	reserved
01	4:2:0
10	reserved
11	reserved

low_delay : This is a one-bit flag which when set to '1' indicates the VOL contains no B-VOPs. If this flag is not present in the bitstream, the default value is 0 for visual object types that support B-VOP otherwise it is 1.

vbv_parameters: This is a one-bit flag which when set to '1' indicates presence of following VBV parameters: first_half_bit_rate, latter_half_bit_rate, first_half_vbv_buffer_size, latter_half_vbv_buffer_size, first_half_vbv_occupancy and latter_half_vbv_occupancy. The VBV constraint is defined in annex D.

first_half_bit_rate, latter_half_bit_rate: The bit rate is a 30-bit unsigned integer which specifies the bitrate of the bitstream measured in units of 400 bits/second, rounded upwards. The value zero is forbidden. This value is divided to two parts. The most significant bits are in first_half_bit_rate (15 bits) and the least significant bits are in latter_half_bit_rate (15 bits). The marker_bit is inserted between the first_half_bit_rate and the latter_half_bit_rate in order to avoid the resync_marker emulation. The instantaneous video object layer channel bit rate seen by the encoder is denoted by $R_{vol}(t)$ in bits per second. If the bit_rate (i.e. first_half_bit_rate and latter_half_bit_rate) field in the VOL header is present, it defines a peak rate (in units of 400 bits per second; a value of 0 is forbidden) such that $R_{vol}(t) <= 400 \times bit\_rate$ Note that $R_{vol}(t)$ counts only visual syntax for the current elementary stream (also see annex D).

first_half_vbv_buffer_size, latter_half_vbv_buffer_size: vbv_buffer_size is an 18-bit unsigned integer. This value is divided into two parts. The most significant bits are in first_half_vbv_buffer_size (15 bits) and the least significant bits are in latter_half_vbv_buffer_size (3 bits), The VBV buffer size is specified in units of 16384 bits. The value 0 for vbv_buffer_size is forbidden. Define B = 16384 × vbv_buffer_size to be the VBV buffer size in bits.

first_half_vbv_occupancy, latter_half_vbv_occupancy: The vbv_occupancy is a 26-bit unsigned integer. This value is divided to two parts. The most significant bits are in first_half_vbv_occupancy (11 bits) and the least significant bits are in latter_half_vbv_occupancy (15 bits). The marker_bit is inserted between the first_half_vbv_occupancy and the latter_half_vbv_occupancy in order to avoid the resync_marker emulation. The value of this integer is the VBV occupancy in 64-bit units just before the removal of the first VOP following the VOL header. The purpose for the quantity is to provide the initial condition for VBV buffer fullness.

video_object_layer_shape: This is a 2-bit integer defined in Table 6-14. It identifies the shape type of a video object layer.

Table 6-14 -- Video Object Layer shape type

Shape format	Meaning
00	rectangular
01	binary
10	binary only
11	grayscale

video_object_layer_shape_extension: This is a 4-bit integer defined in Table V2 - 1. It identifies the number (up to 3) and type of auxiliary components that can be used, including the grayscale shape (ALPHA) component. Only a limited number of types and combinations are defined in Table V2 - 1. More applications are possible by selection of the USER DEFINED type.

Table V2 - 1 -- Semantic meaning of video_object_layer_shape_extension

video_ object_ layer_ shape_ extension	aux_comp_type[0]	aux_comp_type[1]	aux_comp_type[2]	aux_ comp_ count
0000	ALPHA	NO	NO	1
0001	DISPARITY	NO	NO	1
0010	ALPHA	DISPARITY	NO	2
0011	DISPARITY	DISPARITY	NO	2
0100	ALPHA	DISPARITY	DISPARITY	3
0101	DEPTH	NO	NO	1
0110	ALPHA	DEPTH	NO	2
0111	TEXTURE	NO	NO	1
1000	USER DEFINED	NO	NO	1
1001	USER DEFINED	USER DEFINED	NO	2
1010	USER DEFINED	USER DEFINED	USER DEFINED	3
1011	ALPHA	USER DEFINED	NO	2
1100	ALPHA	USER DEFINED	USER DEFINED	3
1101-1111	t.b.d.	t.b.d.	t.b.d.	t.b.d.

vop_time_increment_resolution: This is a 16-bit unsigned integer that indicates the number of evenly spaced subintervals, called ticks, within one modulo time. One modulo time represents the fixed interval of one second. The value zero is forbidden.

fixed_vop_rate: This is a one-bit flag which indicates that all VOPs are coded with a fixed VOP rate. It shall only be '1' if and only if all the distances between the display time of any two successive VOPs in the display order in the video object layer are constant. In this case, the VOP rate can be derived from the fixed_VOP_time_increment. If it is '0' the display time between any two successive VOPs in the display order can be variable thus indicated by the time stamps provided in the VOP header.

fixed_vop_time_increment: This value represents the number of ticks between two successive VOPs in the display order. The length of a tick is given by VOP_time_increment_resolution. It can take a value in the range of [0,VOP_time_increment_resolution). The number of bits representing the value is calculated as the minimum number of unsigned integer bits required to represent the above range. fixed_VOP_time_increment shall only be present if fixed_VOP_rate is '1' and its value must be identical to the constant given by the distance between the display time of any two successive VOPs in the display order. In this case, the fixed VOP rate is given as (VOP_time_increment_resolution / fixed_VOP_time_increment). A zero value is forbidden.

EXAMPLE
VOP time = tick × vop_time_increment
= vop_time_increment / vop_time_increment_resolution

Table 6-15 -- Examples of vop_time_increment_resolution, fix_vop_time_increment, and vop_time_increment

Fixed VOP rate = 1/VOP time	vop_time_increment_ resolution	fixed_vop_time_ increment	vop_time_increment
15Hz	15	1	0, 1, 2, 3, 4,...
7.5Hz	15	2	0, 2, 4, 6, 8,...
29.97...Hz	30000	1001	0, 1001, 2002, 3003,...
59.94...Hz	60000	1001	0, 1001, 2002, 3003,...

video_object_layer_width: The video_object_layer_width is a 13-bit unsigned integer representing the width of the displayable part of the luminance component in pixel units. The width of the encoded luminance component of VOPs in macroblocks is (video_object_layer_width+15)/16. The displayable part is left-aligned in the encoded VOPs. A zero value is forbidden.

video_object_layer_height: The video_object_layer_height is a 13-bit unsigned integer representing the height of the displayable part of the luminance component in pixel units. The height of the encoded luminance component of VOPs in macroblocks is (video_object_layer_height+15)/16. The displayable part is top-aligned in the encoded VOPs. A zero value is forbidden.

interlaced: This is a 1 bit flag which, when set to "1" indicates that the VOP may contain interlaced video. When this flag is set to "0", the VOP is of non-interlaced (or progressive) format.

obmc_disable: This is a one-bit flag which when set to '1' disables overlapped block motion compensation.

sprite_enable: When video_object_layer_verid == '0001', this is a one-bit flag which when set to '1' indicates the usage of static (basic or low latency) sprite coding. When video_object_layer_verid == '0002', this is a two-bit unsigned integer which indicates the usage of static sprite coding or global motion compensation (GMC). Table V2 - 2 shows the meaning of various codewords. An S-VOP with sprite_enable == "GMC" is referred to as an S (GMC)-VOP in this document.

Table V2 - 2 – Meaning of sprite_enable codewords

sprite_enable (video_object_layer_ verid == '0001')	sprite_enable (video_object_layer_ verid == '0002')	Sprite Coding Mode
0	00	sprite not used
1	01	static (Basic/Low Latency)
–	10	GMC (Global Motion Compensation)
–	11	Reserved

sprite_width: This is a 13-bit unsigned integer which identifies the horizontal dimension of the sprite.

sprite_height: This is a 13-bit unsigned integer which identifies the vertical dimension of the sprite.

sprite_left_coordinate: This is a 13-bit signed integer which defines the left edge of the sprite. The value of sprite_left_coordinate shall be divisible by two.

sprite_top_coordinate: This is a 13-bit signed integer which defines the top edge of the sprite. The value of sprite_top_coordinate shall be divisible by two.

no_of_sprite_warping_points: This is a 6-bit unsigned integer which represents the number of points used in sprite warping. When its value is 0 and when sprite_enable is set to 'static' or 'GMC', warping is identity (stationary sprite) and no coordinates need to be coded. When its value is 4, a perspective transform is used. When its value is 1,2 or 3, an affine transform is used. Further, the case of value 1 is separated as a special case from that of values 2 or 3. Table 6-16 shows the various choices. Note that the value of 4 is disallowed when sprite_enable == 'GMC'.

Table 6-16 -- Number of point and implied warping function

Number of points	warping function
0	Stationary
1	Translation
2,3	Affine
4	Perspective
5-63	Reserved

sprite_warping_accuracy – This is a 2-bit code which indicates the quantisation accuracy of motion vectors used in the warping process for sprites and GMC. Table 6-17 shows the meaning of various codewords

Table 6-17 -- Meaning of sprite warping accuracy codewords

code	sprite_warping_accuracy
00	½ pixel
01	¼ pixel
10	1/8 pixel
11	1/16 pixel

sprite_brightness_change: This is a one-bit flag which when set to '1' indicates a change in brightness during sprite warping, alternatively, a value of '0' means no change in brightness.

low_latency_sprite_enable: This is a one-bit flag which when set to "1" indicates the presence of low_latency sprite, alternatively, a value of "0" means basic sprite.

not_8_bit: This one bit flag is set when the video data precision is not 8 bits per pixel and visual object type is N-bit.

sadct_disable: This is a one-bit flag specifying the inverse transforms to be used for texture decoding. If 'sadct_disable' is set to '1', standard inverse DCT as described in version 1 is applied to all 8x8-blocks. When set to '0', flag 'sadct_disable' indicates that different types of inverse DCT are used in an adaptive way: standard inverse DCT is applied to those 8x8-blocks where all 64 pels are opaque, whereas inverse shape-adaptive DCT (SA-DCT) and inverse ΔDC-SA-DCT – an extended version of SA-DCT - are used in inter- and intra-coded 8x8-blocks with at least one transparent and one opaque pel .

quant_precision: This field specifies the number of bits used to represent quantiser parameters. Values between 3 and 9 are allowed. When not_8_bit is zero, and therefore quant_precision is not transmitted, it takes a default value of 5.

bits_per_pixel: This field specifies the video data precision in bits per pixel. It may take different values for different video object layers within a single video object. A value of 12 in this field would indicate 12 bits per pixel. This field may take values between 4 and 12. When not_8_bit is zero and bits_per_pixel is not present, the video data precision is always 8 bits per pixel, which is equivalent to specifying a value of 8 in this field. The same

number of bits per pixel is used in the luminance and two chrominance planes. The alpha plane, used to specify shape of video objects, is always represented with 8 bits per pixel.

no_gray_quant_update: This is a one bit flag which is set to '1' when a fixed quantiser is used for the decoding of grayscale alpha data. When this flag is set to '0', the grayscale alpha quantiser is updated on every macroblock by generating it anew from the luminance quantiser value, but with an appropriate scale factor applied. See the description in subclause 7.5.4.3.

composition_method: This is a one bit flag which indicates which blending method is to be applied to the video object in the compositor. When set to '0', cross-fading shall be used. When set to '1', additive mixing shall be used. See subclause 7.5.4.6.

linear_composition: This is a one bit flag which indicates the type of signal used by the compositing process. When set to '0', the video signal in the format from which it was produced by the video decoder is used. When set to '1', linear signals are used. See subclause 7.5.4.6.

quant_type: This is a one-bit flag which when set to '1' that the first inverse quantisation method and when set to '0' indicates that the second inverse quantisation method is used for inverse quantisation of the DCT coefficients. Both inverse quantisation methods are described in subclause 7.4.4. For the first inverse quantisation method, two matrices are used, one for intra blocks the other for non-intra blocks.

The default matrix for intra blocks is:

8	17	18	19	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	28	30	32	35	38	41
27	28	30	32	35	38	41	45

The default matrix for non-intra blocks is:

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

load_intra_quant_mat: This is a one-bit flag which is set to '1' when intra_quant_mat follows. If it is set to '0' then there is no change in the values that shall be used.

intra_quant_mat: This is a list of 2 to 64 eight-bit unsigned integers. The new values are in zigzag scan order and replace the previous values. A value of 0 indicates that no more values are transmitted and the remaining, non-transmitted values are set equal to the last non-zero value. The first value shall always be 8 and is not used in the decoding process.

load_nonintra_quant_mat: This is a one-bit flag which is set to '1' when nonintra_quant_mat follows. If it is set to '0' then there is no change in the values that shall be used.

nonintra_quant_mat: This is a list of 2 to 64 eight-bit unsigned integers. The new values are in zigzag scan order and replace the previous values. A value of 0 indicates that no more values are transmitted and the remaining, non-transmitted values are set equal to the last non-zero value. The first value shall not be 0.

load_intra_quant_mat_grayscale: This is a one-bit flag which is set to '1' when intra_quant_mat_grayscale follows. If it is set to '0' then there is no change in the quantisation matrix values that shall be used.

intra_quant_mat_grayscale: This is a list of 2 to 64 eight-bit unsigned integers defining the grayscale intra alpha quantisation matrix to be used. The semantics and the default quantisation matrix are identical to those of intra_quant_mat.

load_nonintra_quant_mat_grayscale: This is a one-bit flag which is set to '1' when nonintra_quant_mat_grayscale[i] follows for grayscale alpha or auxiliary component i=0,1,2. If it is set to '0' then there is no change in the quantisation matrix values that shall be used.

intra_quant_mat_grayscale[i]: This is a list of 2 to 64 eight-bit unsigned integers defining the grayscale intra alpha quantisation matrix to be used for grayscale alpha or auxiliary component i=0,1,2. The semantics and the default quantisation matrix are identical to those of intra_quant_mat.

nonintra_quant_mat_grayscale: This is a list of 2 to 64 eight-bit unsigned integers defining the grayscale nonintra alpha quantisation matrix[i] to be used for grayscale alpha or auxiliary component i=0,1,2. The semantics and the default quantisation matrix are identical to those of nonintra_quant_mat.

nonintra_quant_mat_grayscale[i]: This is a list of 2 to 64 eight-bit unsigned integers defining the grayscale nonintra alpha quantisation matrix to be used for grayscale alpha or auxiliary component i=0,1,2. The semantics and the default quantisation matrix are identical to those of nonintra_quant_mat.

quarter_sample: This is a one-bit flag which when set to '0' indicates that half sample mode and when set to '1' indicates that quarter sample mode shall be used for motion compensation of the luminance component.

complexity_estimation_disable: This is a one-bit flag which, when set to '1', disables complexity estimation header in each VOP.

estimation_method: Setting of the of the estimation method, it is „00" for Version 1 and "01" for version 2.

shape_complexity_estimation_disable: This is a one-bit flag which when set to '1' disables shape complexity estimation.

opaque: Flag enabling transmission of the number of luminance and chrominance blocks coded using opaque coding mode in % of the total number of blocks (bounding rectangle).

transparent: Flag enabling transmission of the number of luminance and chrominance blocks coded using transparent mode in % of the total number of blocks (bounding rectangle).

intra_cae: Flag enabling transmission of the number of luminance and chrominance blocks coded using IntraCAE coding mode in % of the total number of blocks (bounding rectangle).

inter_cae: Flag enabling transmission of the number of luminance and chrominance blocks coded using InterCAE coding mode in % of the total number of blocks (bounding rectangle).

no_update: Flag enabling transmission of the number of luminance and chrominance blocks coded using no update coding mode in % of the total number of blocks (bounding rectangle).

upsampling: Flag enabling transmission of the number of luminance and chrominance blocks which need upsampling from 4-4- to 8-8 block dimensions in % of the total number of blocks (bounding rectangle).

version2_complexity_estimation_disable: Flag to disable version 2 parameter set.

sadct: Flag enabling transmission of the number of luminance and chrominance blocks coded using SADCT coding mode in % of the total number of blocks (bounding box). When estimation_method == '00' the value of sadct is set to '0'.

quarterpel: Flag enabling transmission of the number of luminance and chrominance block predicted by a quarter-pel vector on one or two dimensions (horizontal and vertical) in % of the total number of blocks (bounding box). When estimation_method == '00' the value of quarterpel is set to '0'.

texture_complexity_estimation_set_1_disable: Flag to disable texture parameter set 1.

intra_blocks: Flag enabling transmission of the number of luminance and chrominance Intra or Intra+Q coded blocks in % of the total number of blocks (bounding rectangle).

inter_blocks: Flag enabling transmission of the number of luminance and chrominance Inter and Inter+Q coded blocks in % of the total number of blocks (bounding rectangle).

inter4v_blocks: Flag enabling transmission of the number of luminance and chrominance Inter4V coded blocks in % of the total number of blocks (bounding rectangle).

not_coded_blocks: Flag enabling transmission of the number of luminance and chrominance Non Coded blocks in % of the total number of blocks (bounding rectangle).

texture_complexity_estimation_set_2_disable: Flag to disable texture parameter set 2.

dct_coefs: Flag enabling transmission of the number of DCT coefficients % of the maximum number of coefficients (coded blocks).

dct_lines: Flag enabling transmission of the number of DCT8x1 in % of the maximum number of DCT8x1 (coded blocks).

vlc_symbols: Flag enabling transmission of the average number of VLC symbols for macroblock.

vlc_bits: Flag enabling transmission of the average number of bits for each symbol.

motion_compensation_complexity_disable: Flag to disable motion compensation parameter set.

apm (Advanced Prediction Mode): Flag enabling transmission of the number of luminance block predicted using APM in % of the total number of blocks for VOP (bounding rectangle).

npm (Normal Prediction Mode): Flag enabling transmission of the number of luminance and chrominance blocks predicted using NPM in % of the total number of luminance and chrominance for VOP (bounding rectangle).

interpolate_mc_q: Flag enabling transmission of the number of luminance and chrominance interpolated blocks in % of the total number of blocks for VOP (bounding rectangle).

forw_back_mc_q: Flag enabling transmission of the number of luminance and chrominance predicted blocks in % of the total number of blocks for VOP (bounding rectangle).

halfpel2: Flag enabling transmission of the number of luminance and chrominance block predicted by a half-pel vector on one dimension (horizontal or vertical) in % of the total number of blocks (bounding rectangle).

halfpel4: Flag enabling transmission of the number of luminance and chrominance block predicted by a half-pel vector on two dimensions (horizontal and vertical) in % of the total number of blocks (bounding rectangle).

resync_marker_disable: This is a one-bit flag which when set to '1' indicates that there is no resync_marker in coded VOPs. This flag can be used only for the optimization of the decoder operation. Successful decoding can be carried out without taking into account the value of this flag.

data_partitioned: This is a one-bit flag which when set to '1' indicates that the macroblock data is rearranged differently, specifically, motion vector data is separated from the texture data (i.e., DCT coefficients).

reversible_vlc: This is a one-bit flag which when set to '1' indicates that the reversible variable length tables (Table B-23, Table B-24 and Table B-25) should be used when decoding DCT coefficients. These tables can only be used when data_partition flag is enabled. Note that this flag shall be treated as '0' in B-VOPs. Use of escape sequence (Table B-24 and Table B-25) for encoding the combinations listed in Table B-23 is prohibited.

newpred_enable: This is a one-bit flag which, when set to '1' ,indicates that the NEWPRED mode is enabled. When video_object_layer_verid is equal to '0001', and therefore newpred enable is not transmitted, it takes a default value of zero.

requested_upstream_message_type: This is a twe-bits flag which indicates which type of upstream message is needed by the encoder. The syntax and semantics of the upstream message are described in subclause 6.2.12 and 6.3.12.

01: need NP_ACK message to be returned for each NEWPRED segment
10: need NP_NACK message to be returned for each NEWPRED segment
11: need both NP_ACK and NP_NACK messages to be returned for each NEWPRED segment
00: reserved

newpred_segment_type: This is a one-bits flag which indicates the unit of selecting reference VOP (NEWPRED segment).

0: Video Packet
1: VOP

reduced_resolution_vop_enable: This is a one-bit flag which indicates that the reduced resolution vop tool is enabled when set to '1'. When video_object_layer_verid is equal to '0001', and therefore reduced_resolution_vop_enable is not transmitted, it takes a default value of zero.

scalability: This is a one-bit flag which when set to '1' indicates that the current layer uses scalable coding. If the current layer is used as base-layer then this flag is set to '0'. Additionally, this flag shall be set to '0' for S(GMC)-VOPs.

hierarchy_type: The hierarcical relation between the associated hierarchy layer and its hierarchy embedded layer is defined as shown in Table 6-18.

Table 6-18 -- Code table for hierarchy_type

Description	Code
ISO/IEC 14496-2 Spatial Scalability	0
ISO/IEC 14496-2 Temporal Scalability	1

ref_layer_id: This is a 4-bit unsigned integer with value between 0 and 15. It indicates the layer to be used as reference for prediction(s) in the case of scalability.

ref_layer_sampling_direc: This is a one-bit flag which when set to '1' indicates that the resolution of the reference layer (specified by reference_layer_id) is higher than the resolution of the layer being coded. If it is set to '0' then the reference layer has the same or lower resolution than the resolution of the layer being coded.

hor_sampling_factor_n: This is a 5-bit unsigned integer which forms the numerator of the ratio used in horizontal spatial resampling in scalability. The value of zero is forbidden.

hor_sampling_factor_m: This is a 5-bit unsigned integer which forms the denominator of the ratio used in horizontal spatial resampling in scalability. The value of zero is forbidden.

vert_sampling_factor_n: This is a 5-bit unsigned integer which forms the numerator of the ratio used in vertical spatial resampling in scalability. The value of zero is forbidden.

vert_sampling_factor_m: This is a 5-bit unsigned integer which forms the denominator of the ratio used in vertical spatial resampling in scalability. The value of zero is forbidden.

enhancement_type: This is a 1-bit flag which is set to '1' when the current layer enhances the partial region of the reference layer. If it is set to '0' then the current layer enhances the entire region of the reference layer. The default value of this flag is '0'.

use_ref_shape: This is one bit flag which indicate procedure to decode binary shape for spatial scalability. If it is set to '0', scalable shape coding should be used. If it is set to '1' and enhancement_type is set to '0', no shape data is decoded and up-sampled binary shape of base layer should be used for enhancement layer. If enhancement_type is set to '1' and this flag is set to '1',binary shape of enhancement layer should be decoded as the same non-scalable decoding process. When video_object_layer_verid == '0001', the value of use_ref_shape_ is set to '1'.

use_ref_texture: When this one bit is set, no update for texture is done. Instead, the available texture in the layer denoted by ref_layer_id will be used.

shape_hor_sampling_factor_n: This is a 5-bit unsigned integer which forms the numerator of the ratio used in horizontal spatial resampling in shape scalability. The value of zero is forbidden.

shape_hor_sampling_factor_m: This is a 5-bit unsigned integer which forms the denominator of the ratio used in horizontal spatial resampling in shape scalability. The value of zero is forbidden.

shape_vert_sampling_factor_n: This is a 5-bit unsigned integer which forms the denominator of the ratio used in vertical spatial resampling in shape scalability. The value of zero is forbidden.

shape_vert_sampling_factor_m: This is a 5-bit unsigned integer which forms the denominator of the ratio used in vertical spatial resampling in shape scalability. The value of zero is forbidden.

6.3.4 Group of Video Object Plane

group_of_vop_start_code: The group_of_vop_start_code is the bit string '000001B3' in hexadecimal. It identifies the beginning of a GOV header.

time_code: This is a 18-bit integer containing the following: time_code_hours, time_code_minutes, marker_bit and time_code_seconds as shown in Table 6-19. The parameters correspond to those defined in the IEC standard publication 461 for "time and control codes for video tape recorders". The time code specifies the modulo part (i.e. the full second units) of the time base for the first object plane (in display order) after the GOV header.

Table 6-19 -- Meaning of time_code

time_code	range of value	No. of bits	Mnemonic
time_code_hours	0 - 23	5	uimsbf
time_code_minutes	0 - 59	6	uimsbf
marker_bit	1	1	bslbf
time_code_seconds	0 - 59	6	uimsbf